IPv6 Extension Headers (PDM) and Testing on the Internet

Presented by: Nalini Elkins

NALINI ELKINS

INDUSTRY NETWORK TECHNOLOGY COUNCIL

PRESIDENT@INDUSTRYNETCOUNCIL.ORG

PRANEET KAUR

INDIA INTERNET ENGINEERING SOCIETY

INFO@IIESOC.IN

A few words about me

- President: Industry Network Technology Council
- Founder & CEO: Inside Products, Inc.
- Advisory Board: India Internet Engineering Society
- RFCs: RFC8250 (Embedded performance and diagnostics for IPv6) and others
- Product developer (OEMed by IBM and others)
- Working with IPv6 for 15 years
- Working with network management, diagnostic, performance issues at large brick-and-mortar enterprises for over 30 years

Thanks to…

IIESoc

IIESoc

India Internet Engineering Society

National Institute of Technology Karnataka, Surathkal

राष्ट्रीय प्रौद्योगिकी संस्थान कर्नाटक, सुरत्कल

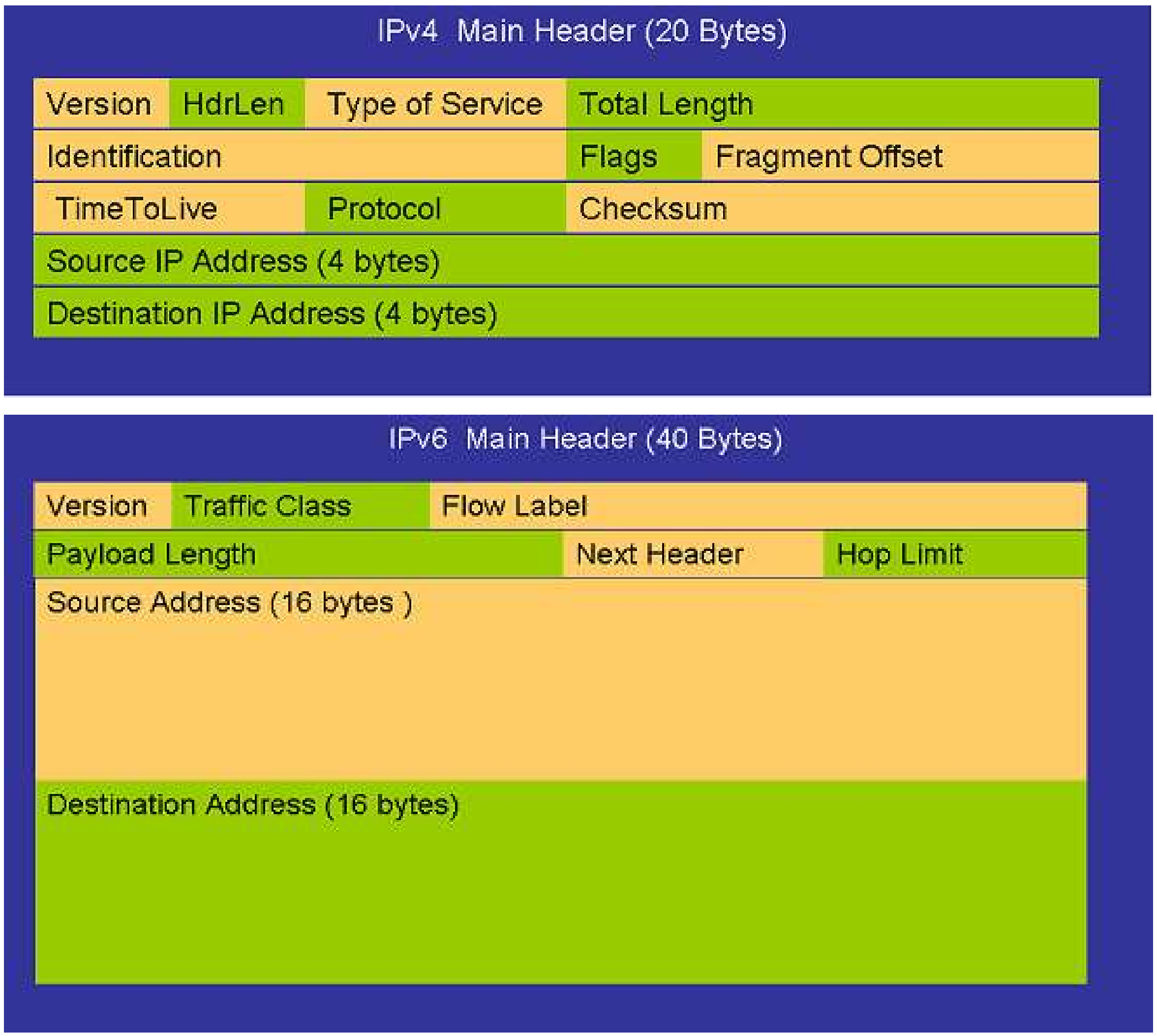
In particular, Dr. Mohit Tahiliani

Industry Network Technology Council

Agenda

- IPv6 address structure
- IPv6 extension headers
- EH testing: standalone
- EH testing: CDN
- NITK Student presentation

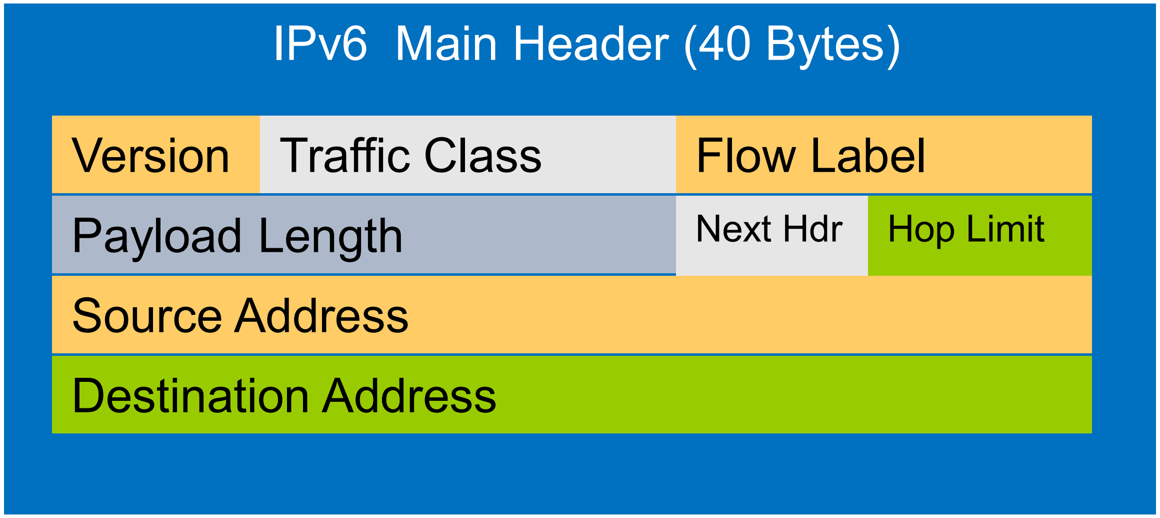
IPv4 and IPv6 Headers



- What is the same?
- What is different?

The IPv6 Header

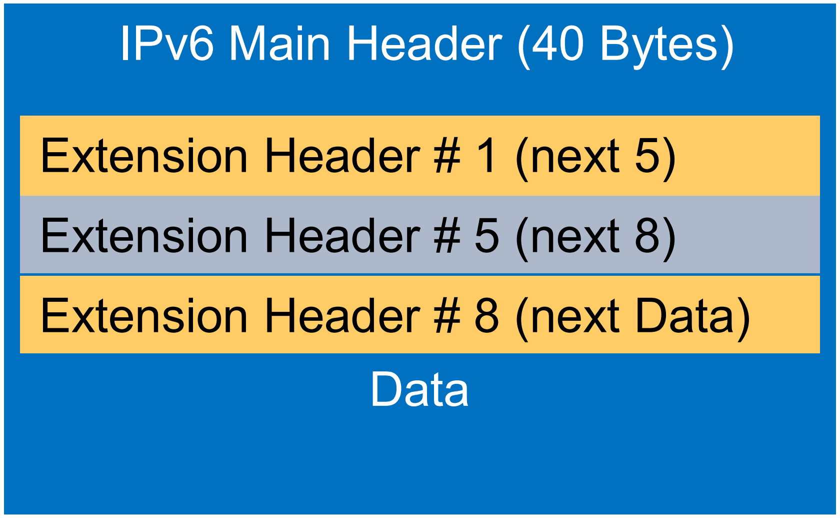
- IPv6 main header: fixed 40 bytes
- Source and destination addresses larger!
- Defined in RFC8200 (originally RFC2460)



IPv6 Extension Headers

- New: IPv6 extension headers
- Next Header field chains headers

- Rules:
  - May appear only once
  - Must appear in fixed order
  - Exception: Destination Options



Common IPv6 Extension Headers

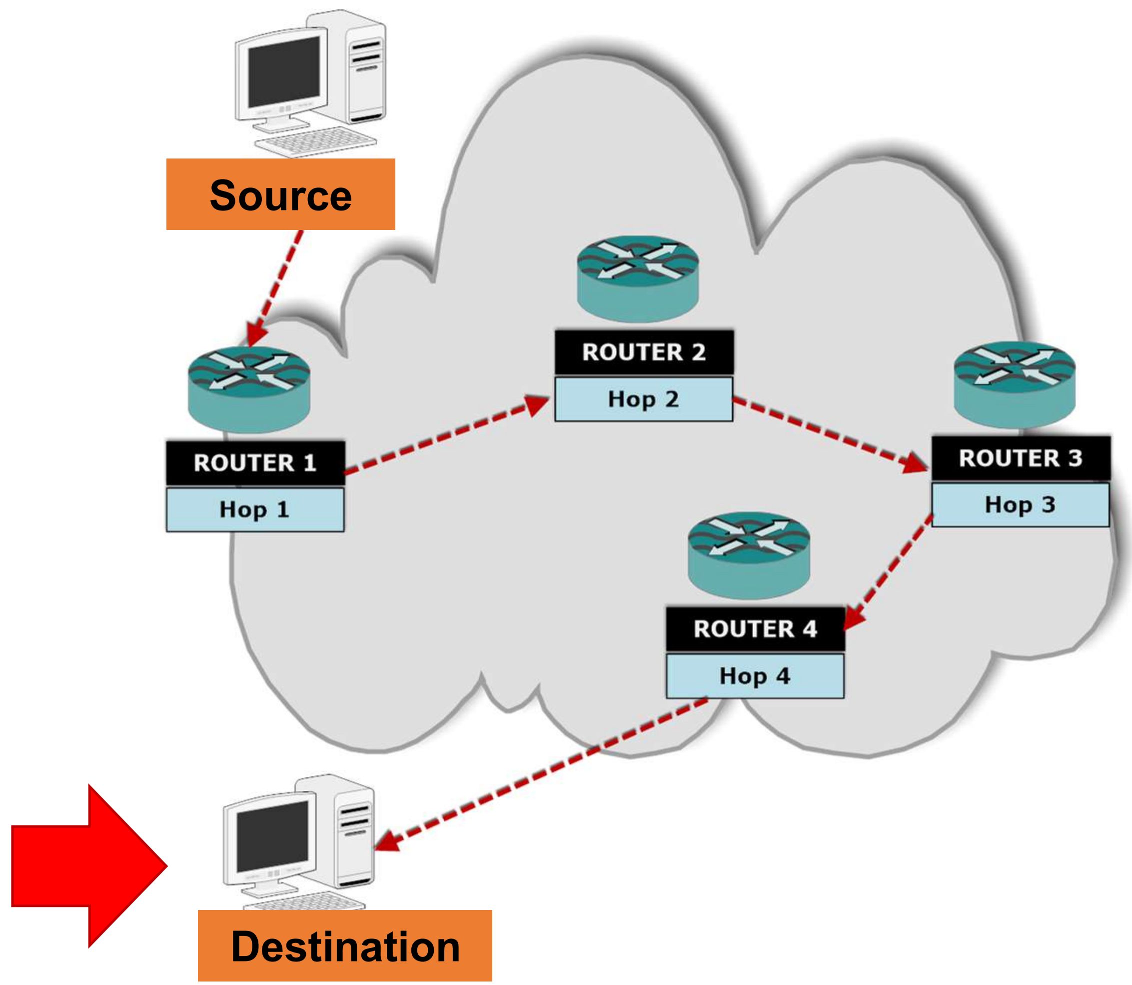
Next Header (Hex)	Next Header (Decimal)	Header Name	Description
0	0	Hop-by-Hop Options	For all devices on the path
2B	43	Routing	0 – Source Routing (deprecated) 2 – Mobile IPv6
2C	44	Fragment	Only when packet is fragmented
32	50	Encapsulated Security Payload (ESP)	IPSec encrypted data
33	51	Authentication Header (AH)	IPSec authentication
3C	60	Destination Options	http://www.iana.org/assignments/ipv6-parameters/ipv6-parameters.xml (Mobile IP, etc)

No.	Time	Source	Destination	Pro
1693	46.130640	::	ff02::2	IC

```
+ Frame 1693 (86 bytes on wire, 86 bytes captured)
- Ethernet II, Src: 192.168.1.1 (00:14:bf:ba:45:f9), Dst: I
    Destination: IPv6-Neighbor-Discovery_00:00:00:02 (33:33
    Source: 192.168.1.1 (00:14:bf:ba:45:f9)
    Type: IPv6 (0x86dd)
- Internet Protocol Version 6
    Version: 6
    Traffic class: 0x00
    Flowlabel: 0x00000
    Payload length: 32
    Next header: IPv6 hop-by-hop option (0x00)
    Hop limit: 1
    Source address: ::
    Destination address: ff02::2
- Hop-by-hop Option Header
    Next header: ICMPv6 (0x3a)
    Length: 0 (8 bytes)
    Router alert: MLD (4 bytes)
    PadN: 2 bytes
- Internet Control Message Protocol v6
    Type: 131 (Multicast listener report)
    Code: 0
    Checksum: 0x7ea3 [correct]
    Maximum response delay: 0
    Multicast Address: ff02::2
```

IPv6 Destination Options

- Destination Options: for end host



IPv6 Destination Options

```
⊞ Frame 1: 260 bytes on wire (2080 bits), 260 bytes captured (2080 bits)
⊞ Prism capture header
⊞ IEEE 802.11 Data, Flags: .......T
⊞ Logical-Link Control
⊟ Internet Protocol Version 6, Src: 2001:720:810:1212:209:b7ff:fe3c:902c (2001:720:810:1212:209:b7
  ⊞ 0110 .... = Version: 6
  ⊞ .... 0000 0000 .... .... .... .... .... = Traffic class: 0x00000000
    .... .... .... 0000 0000 0000 0000 0000 = Flowlabel: 0x00000000
    Payload length: 40
    Next header: IPv6 destination option (60)
    Hop limit: 255
    Source: 2001:720:810:1212:209:b7ff:fe3c:902c (2001:720:810:1212:209:b7ff:fe3c:902c)
    [Source SA MAC: Cisco_3c:90:2c (00:09:b7:3c:90:2c)]
    Destination: 2001:720:810:1213::1 (2001:720:810:1213::1)
    [Source GeoIP: Unknown]
    [Destination GeoIP: Unknown]
  ⊟ Destination Option
      Next header: Mobile IPv6 ( (62)
      Length: 2 (24 bytes)
    ⊟ IPv6 Option (PadN)
        Type: PadN (1)
        Length: 2
        PadN: 0000
    ⊟ IPv6 Option (Home Address)
        Type: Home Address (201)
        Length: 16
        Home Address: 2001:720:810:1213::2 (2001:720:810:1213::2)
⊟ Mobile IPv6 / Network Mobility
```

Use of Destination Options in Mobile IPv6

Issues

- Routers may drop
- Extension headers may be too big (ASIC size)
- Privacy violations
- Change the extension header not at source
- Parsing of TLV (Type, length, value)

Pros

- New protocol functions = extensible protocol, long maturity
- PDM / PDMv2 : embedded performance and diagnostic metrics + encryption methodology for IPv6 extension headers (RFC8250, et al)
- Many others: source address validation

Can IPv6 Extension Headers Be Used on the Internet?

- Controversy for many years
- A number of studies showing that IPv6 extension headers “don’t work”
- Studies (by and large) sent “fake” IPv6 extension headers to Alexa top n sites
- If this is true, our work on our IPv6 Extension Header Destination Option Performance and Diagnostic Metrics (PDM) is really for naught

Brief explanation of PDM

- RFC8250: IPv6 Performance and Diagnostic Metrics (PDM) Destination Option

- To assess performance problems, this document describes optional headers embedded in each packet that provide sequence numbers and timing information as a basis for measurements. Such measurements may be interpreted in real time or after the fact. This document specifies the Performance and Diagnostic Metrics (PDM) Destination Options header.

. Our Testing Components

We have:

- a test server enabled to send EH with every packet
- an IPv6 enabled web server (Apache)
- a packet trace capture tool such as TCPDump, WireShark, etc.

What we did

- Used a small hosting service (not one of the “brand-name” ones)
- Locations throughout the world

1. PDM-Warsaw
2. PDM-Toronto
3. PDM-Seattle
4. PDM-Mumbai
5. PDM-Melbourne
6. PDM-Frankfurt

All machines are FreeBSD with a modification to the kernel to send PDM IPv6 Destination option with every packet.

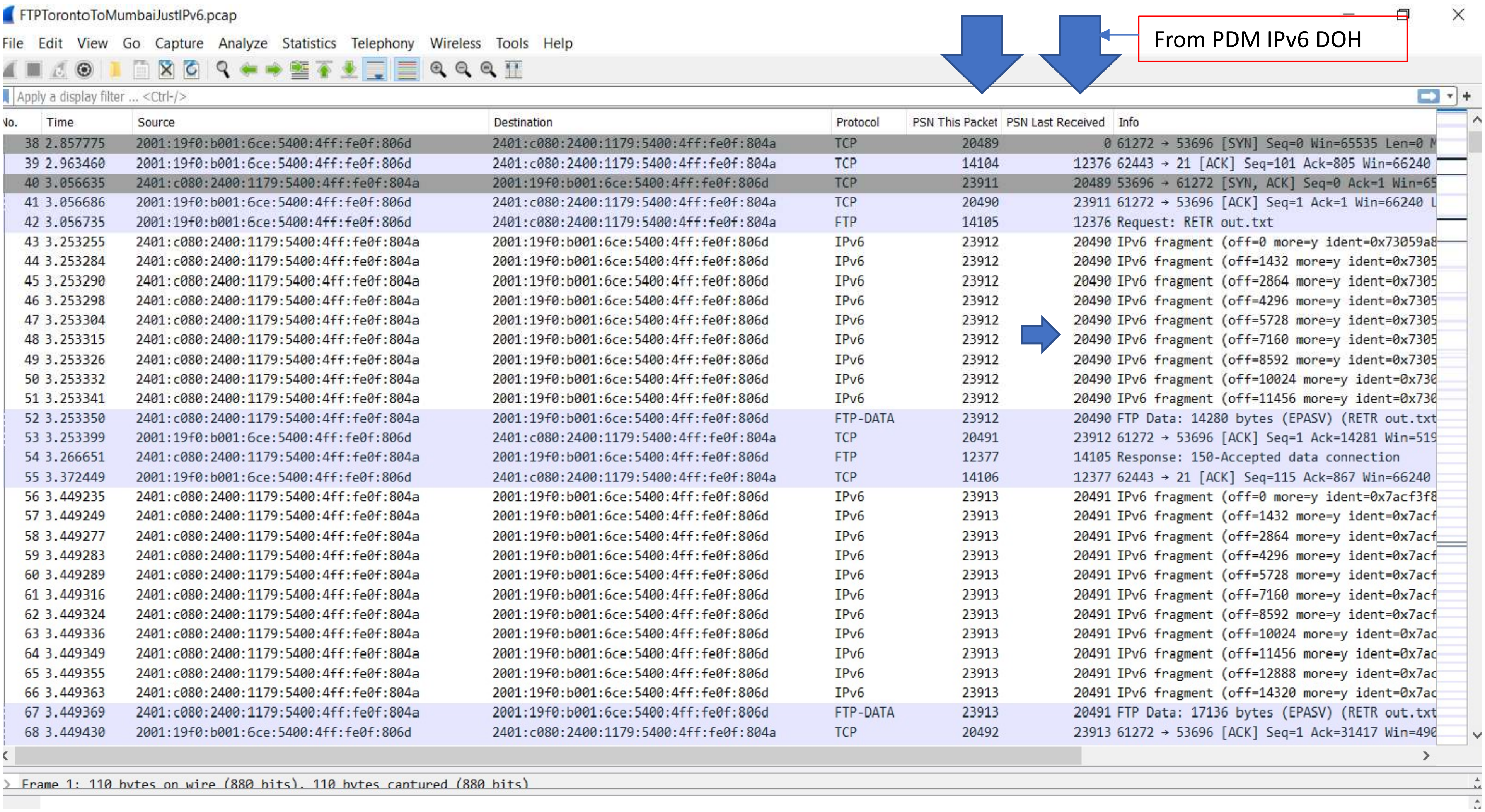
We have changed to use eBPF. NITK students will talk about this.

Tested large FTP: Toronto to Mumbai (with PDM)

- Connected to **2401:c080:2400:1179:5400:04ff:fe0f:804a.**
- 220---------- Welcome to Pure-FTPd [privsep] [TLS] ----------
- 220-You are user number 1 of 50 allowed.
- 220-Local time is now 15:12. Server port: 21.
- 220 You will be disconnected after 15 minutes of inactivity.
- 331 User PDMuser OK. Password required
- 230 OK. Current directory is /
- Remote system type is UNIX.
- Using binary mode to transfer files.

- 229 Extended Passive mode OK (|||3353|)
- 150-Accepted data connection
- 150 **27872.0 kbytes to download**
- 100% |**| 27872 KiB 222.31 KiB/s 00:00 ETA
- 226-**File successfully transferred**
- 226 125.107 seconds (measured here), 222.78 Kbytes per second
- 28540928 bytes received in 02:05 (222.31 KiB/s)
- 221-Goodbye. You uploaded 0 and downloaded 27872 kbytes.
- 221 Logout.

Trace of Extension Headers



No.	Time	Source	Destination	Protocol	PSN This Packet	PSN Last Received	Info
38	2.857775	2001:19f0:b001:6ce:5400:4ff:fe0f:806d	2401:c080:2400:1179:5400:4ff:fe0f:804a	TCP	20489	0	61272 → 53696 [SYN] Seq=0 Win=65535 Len=0 M
39	2.963460	2001:19f0:b001:6ce:5400:4ff:fe0f:806d	2401:c080:2400:1179:5400:4ff:fe0f:804a	TCP	14104	12376	62443 → 21 [ACK] Seq=101 Ack=805 Win=66240
40	3.056635	2401:c080:2400:1179:5400:4ff:fe0f:804a	2001:19f0:b001:6ce:5400:4ff:fe0f:806d	TCP	23911	20489	53696 → 61272 [SYN, ACK] Seq=0 Ack=1 Win=65
41	3.056686	2001:19f0:b001:6ce:5400:4ff:fe0f:806d	2401:c080:2400:1179:5400:4ff:fe0f:804a	TCP	20490	23911	61272 → 53696 [ACK] Seq=1 Ack=1 Win=66240 L
42	3.056735	2001:19f0:b001:6ce:5400:4ff:fe0f:806d	2401:c080:2400:1179:5400:4ff:fe0f:804a	FTP	14105	12376	Request: RETR out.txt
43	3.253255	2401:c080:2400:1179:5400:4ff:fe0f:804a	2001:19f0:b001:6ce:5400:4ff:fe0f:806d	IPv6	23912	20490	IPv6 fragment (off=0 more=y ident=0x73059a8
44	3.253284	2401:c080:2400:1179:5400:4ff:fe0f:804a	2001:19f0:b001:6ce:5400:4ff:fe0f:806d	IPv6	23912	20490	IPv6 fragment (off=1432 more=y ident=0x7305
45	3.253290	2401:c080:2400:1179:5400:4ff:fe0f:804a	2001:19f0:b001:6ce:5400:4ff:fe0f:806d	IPv6	23912	20490	IPv6 fragment (off=2864 more=y ident=0x7305
46	3.253298	2401:c080:2400:1179:5400:4ff:fe0f:804a	2001:19f0:b001:6ce:5400:4ff:fe0f:806d	IPv6	23912	20490	IPv6 fragment (off=4296 more=y ident=0x7305
47	3.253304	2401:c080:2400:1179:5400:4ff:fe0f:804a	2001:19f0:b001:6ce:5400:4ff:fe0f:806d	IPv6	23912	20490	IPv6 fragment (off=5728 more=y ident=0x7305
48	3.253315	2401:c080:2400:1179:5400:4ff:fe0f:804a	2001:19f0:b001:6ce:5400:4ff:fe0f:806d	IPv6	23912	20490	IPv6 fragment (off=7160 more=y ident=0x7305
49	3.253326	2401:c080:2400:1179:5400:4ff:fe0f:804a	2001:19f0:b001:6ce:5400:4ff:fe0f:806d	IPv6	23912	20490	IPv6 fragment (off=8592 more=y ident=0x7305
50	3.253332	2401:c080:2400:1179:5400:4ff:fe0f:804a	2001:19f0:b001:6ce:5400:4ff:fe0f:806d	IPv6	23912	20490	IPv6 fragment (off=10024 more=y ident=0x730
51	3.253341	2401:c080:2400:1179:5400:4ff:fe0f:804a	2001:19f0:b001:6ce:5400:4ff:fe0f:806d	IPv6	23912	20490	IPv6 fragment (off=11456 more=y ident=0x730
52	3.253350	2401:c080:2400:1179:5400:4ff:fe0f:804a	2001:19f0:b001:6ce:5400:4ff:fe0f:806d	FTP-DATA	23912	20490	FTP Data: 14280 bytes (EPASV) (RETR out.txt
53	3.253399	2001:19f0:b001:6ce:5400:4ff:fe0f:806d	2401:c080:2400:1179:5400:4ff:fe0f:804a	TCP	20491	23912	61272 → 53696 [ACK] Seq=1 Ack=14281 Win=519
54	3.266651	2401:c080:2400:1179:5400:4ff:fe0f:804a	2001:19f0:b001:6ce:5400:4ff:fe0f:806d	FTP	12377	14105	Response: 150-Accepted data connection
55	3.372449	2001:19f0:b001:6ce:5400:4ff:fe0f:806d	2401:c080:2400:1179:5400:4ff:fe0f:804a	TCP	14106	12377	62443 → 21 [ACK] Seq=115 Ack=867 Win=66240
56	3.449235	2401:c080:2400:1179:5400:4ff:fe0f:804a	2001:19f0:b001:6ce:5400:4ff:fe0f:806d	IPv6	23913	20491	IPv6 fragment (off=0 more=y ident=0x7acf3f8
57	3.449249	2401:c080:2400:1179:5400:4ff:fe0f:804a	2001:19f0:b001:6ce:5400:4ff:fe0f:806d	IPv6	23913	20491	IPv6 fragment (off=1432 more=y ident=0x7acf
58	3.449277	2401:c080:2400:1179:5400:4ff:fe0f:804a	2001:19f0:b001:6ce:5400:4ff:fe0f:806d	IPv6	23913	20491	IPv6 fragment (off=2864 more=y ident=0x7acf
59	3.449283	2401:c080:2400:1179:5400:4ff:fe0f:804a	2001:19f0:b001:6ce:5400:4ff:fe0f:806d	IPv6	23913	20491	IPv6 fragment (off=4296 more=y ident=0x7acf
60	3.449289	2401:c080:2400:1179:5400:4ff:fe0f:804a	2001:19f0:b001:6ce:5400:4ff:fe0f:806d	IPv6	23913	20491	IPv6 fragment (off=5728 more=y ident=0x7acf
61	3.449316	2401:c080:2400:1179:5400:4ff:fe0f:804a	2001:19f0:b001:6ce:5400:4ff:fe0f:806d	IPv6	23913	20491	IPv6 fragment (off=7160 more=y ident=0x7acf
62	3.449324	2401:c080:2400:1179:5400:4ff:fe0f:804a	2001:19f0:b001:6ce:5400:4ff:fe0f:806d	IPv6	23913	20491	IPv6 fragment (off=8592 more=y ident=0x7acf
63	3.449336	2401:c080:2400:1179:5400:4ff:fe0f:804a	2001:19f0:b001:6ce:5400:4ff:fe0f:806d	IPv6	23913	20491	IPv6 fragment (off=10024 more=y ident=0x7ac
64	3.449349	2401:c080:2400:1179:5400:4ff:fe0f:804a	2001:19f0:b001:6ce:5400:4ff:fe0f:806d	IPv6	23913	20491	IPv6 fragment (off=11456 more=y ident=0x7ac
65	3.449355	2401:c080:2400:1179:5400:4ff:fe0f:804a	2001:19f0:b001:6ce:5400:4ff:fe0f:806d	IPv6	23913	20491	IPv6 fragment (off=12888 more=y ident=0x7ac
66	3.449363	2401:c080:2400:1179:5400:4ff:fe0f:804a	2001:19f0:b001:6ce:5400:4ff:fe0f:806d	IPv6	23913	20491	IPv6 fragment (off=14320 more=y ident=0x7ac
67	3.449369	2401:c080:2400:1179:5400:4ff:fe0f:804a	2001:19f0:b001:6ce:5400:4ff:fe0f:806d	FTP-DATA	23913	20491	FTP Data: 17136 bytes (EPASV) (RETR out.txt
68	3.449430	2001:19f0:b001:6ce:5400:4ff:fe0f:806d	2401:c080:2400:1179:5400:4ff:fe0f:804a	TCP	20492	23913	61272 → 53696 [ACK] Seq=1 Ack=31417 Win=490

Showing both Extension Headers

```
v Destination Options for IPv6
    Next Header: Fragment Header for IPv6 (44)
    Length: 1
    [Length: 16 bytes]
  v Performance and Diagnostic Metrics
    > Type: Performance and Diagnostic Metrics (0x0f)
      Length: 10
      Scale DTLR: 34
      Scale DTLS: 42
      PSN This Packet: 23912
      PSN Last Received: 20490
      Delta Time Last Received: 37754
      Delta Time Last Sent: 45216
  v PadN
    > Type: PadN (0x01)
      Length: 0
      PadN: <none>
v Fragment Header for IPv6
    Next header: TCP (6)
    Reserved octet: 0x00
    0000 0000 0000 0... = Offset: 0 (0 bytes)
    .... .... .... .00. = Reserved bits: 0
    .... .... .... ...1 = More Fragments: Yes
    Identification: 0x73059a89
  [Reassembled IPv6 in frame: 52]
> Data (1432 bytes)
```

Bottom line

1. PDM-FTP Toronto to Warsaw - worked
2. PDM-FTP Toronto to Seattle - worked
3. PDM-FTP Toronto to Mumbai - worked
4. PDM-FTP Toronto to Melbourne - worked
5. PDM-FTP Toronto to Frankfurt - worked

Traces available for all to look at.

Is hosting service using an overlay network?

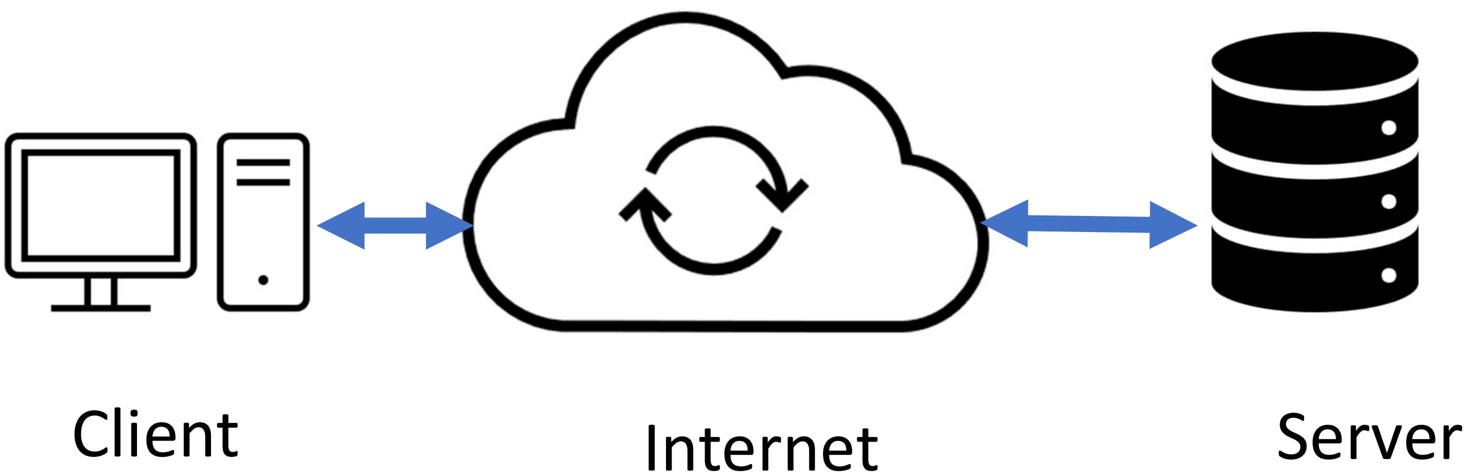
- Email sent by me
  - I have a question about the connection between various [hostingcompany] instances. For example, if I have an instance in Mumbai and another one in Atlanta, then do you have an overlay network? That is, do you have special connectivity between [hostingcompany] instances or is it going over the Internet?

- Response from hosting company
  - Communication between [hostingcompany] VPS residing in different datacenters will always travel on public internet exchanges. [hostingcompany] Private Cloud can create a private network, however this is only for communication between instances in the same datacenter. [hostingcompany] utilizes multiple transit providers.

Why are our results so different from others?

- We are using real data and a real application (e.g. PDM and FTP)
- We are NOT going to the Alexa top n
- But, we also tried to replicate the results of others
- Indeed, if you use the large hosting companies and go to the Alexa top n, there are issues
- But why?

The topology that worked
Simplest: Client – Internet -- Server

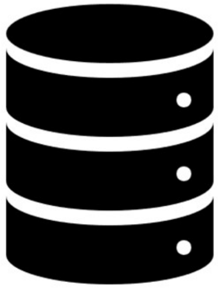


Goal of CDN Testing

- Why look at CDNs?
  - Many high usage websites on the internet use CDNs
  - They have a disproportionate impact on IPv6 and EH use
- Need to figure out
  - Where EH can be sent with 90%+ probability (**and why**)
  - Where EH CANNOT be sent with 90%+ probability (**and why**)
  - What is unknown

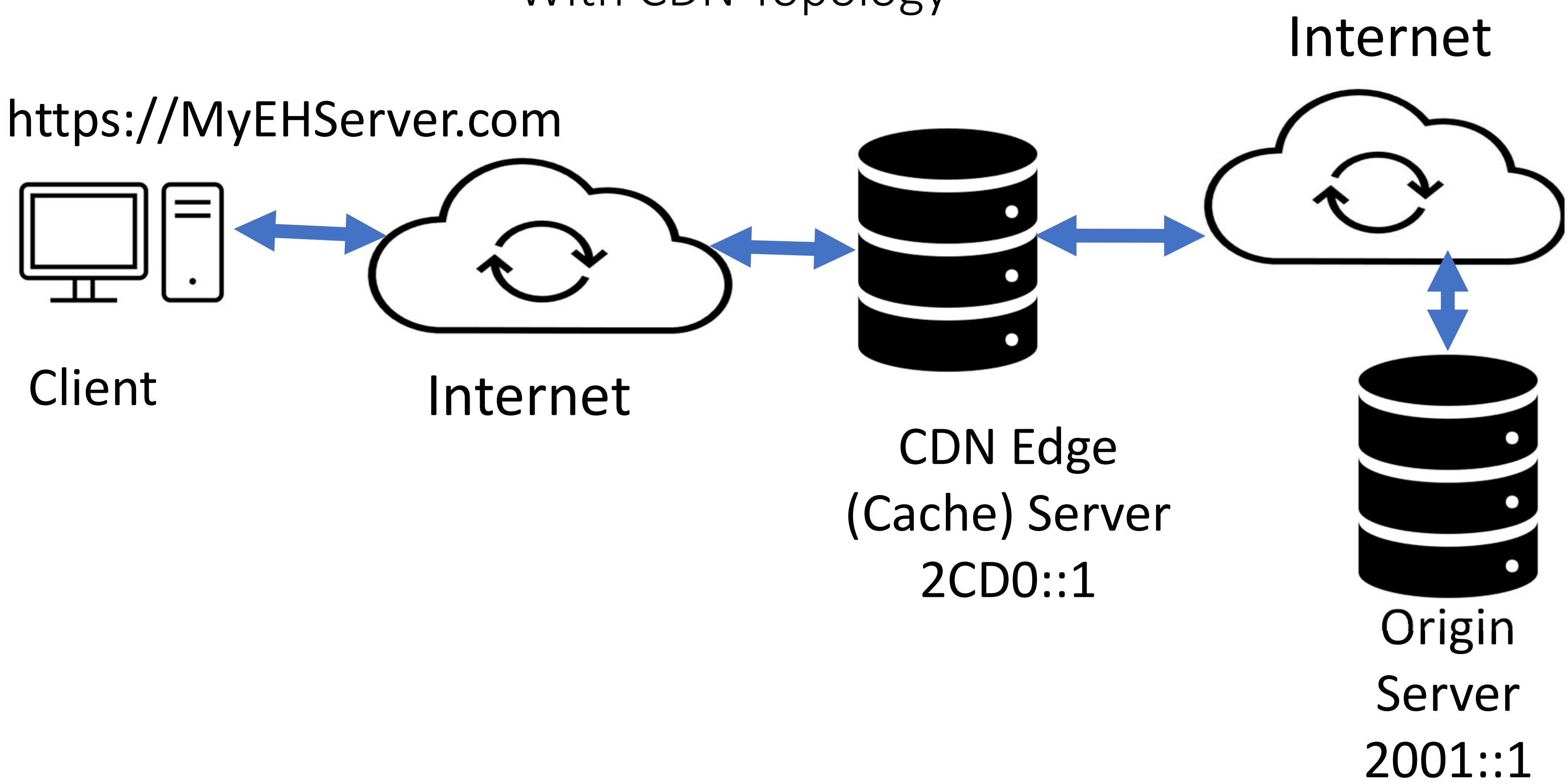
Move Server Behind CDN

- Our server has a domain name: MyEHServer
- Our server also has an IPv6 address (also IPv4 probably)
- Let's say: 2001::1 and 201.1.1.1 (MyEHServer resolves to these)
- To move behind a CDN, you have to give the CDN authority to resolve MyEHServer
- Let's give the CDN IPv6 addresses starting with 2CD0::/64 (2CD0::1, 2CD0::2, etc)
- After CDN move, MyEHServer will resolve to some CDN cache server address (2CD0::1 for example)



We will now refer to our server as the "Origin Server"

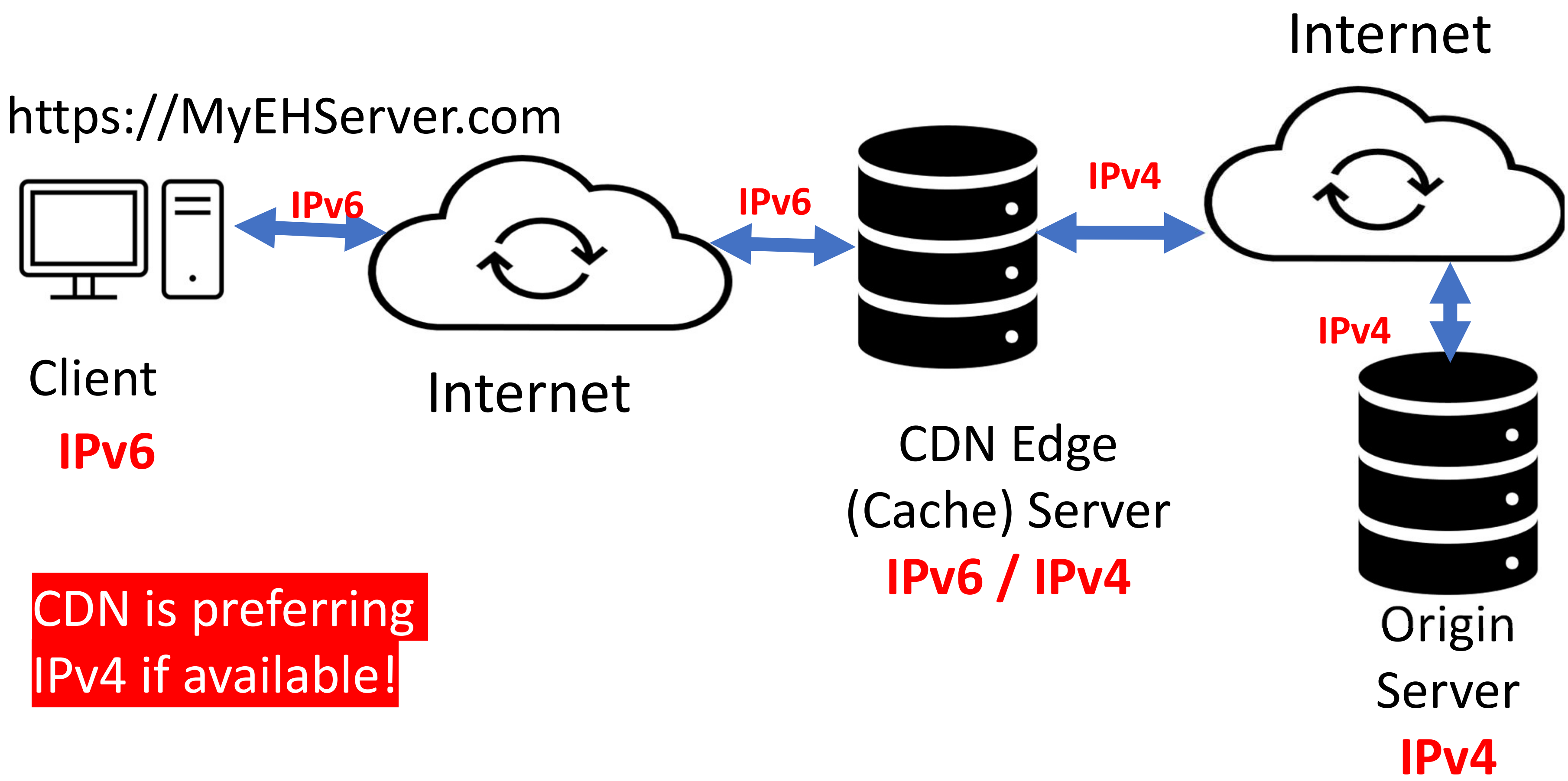
With CDN Topology



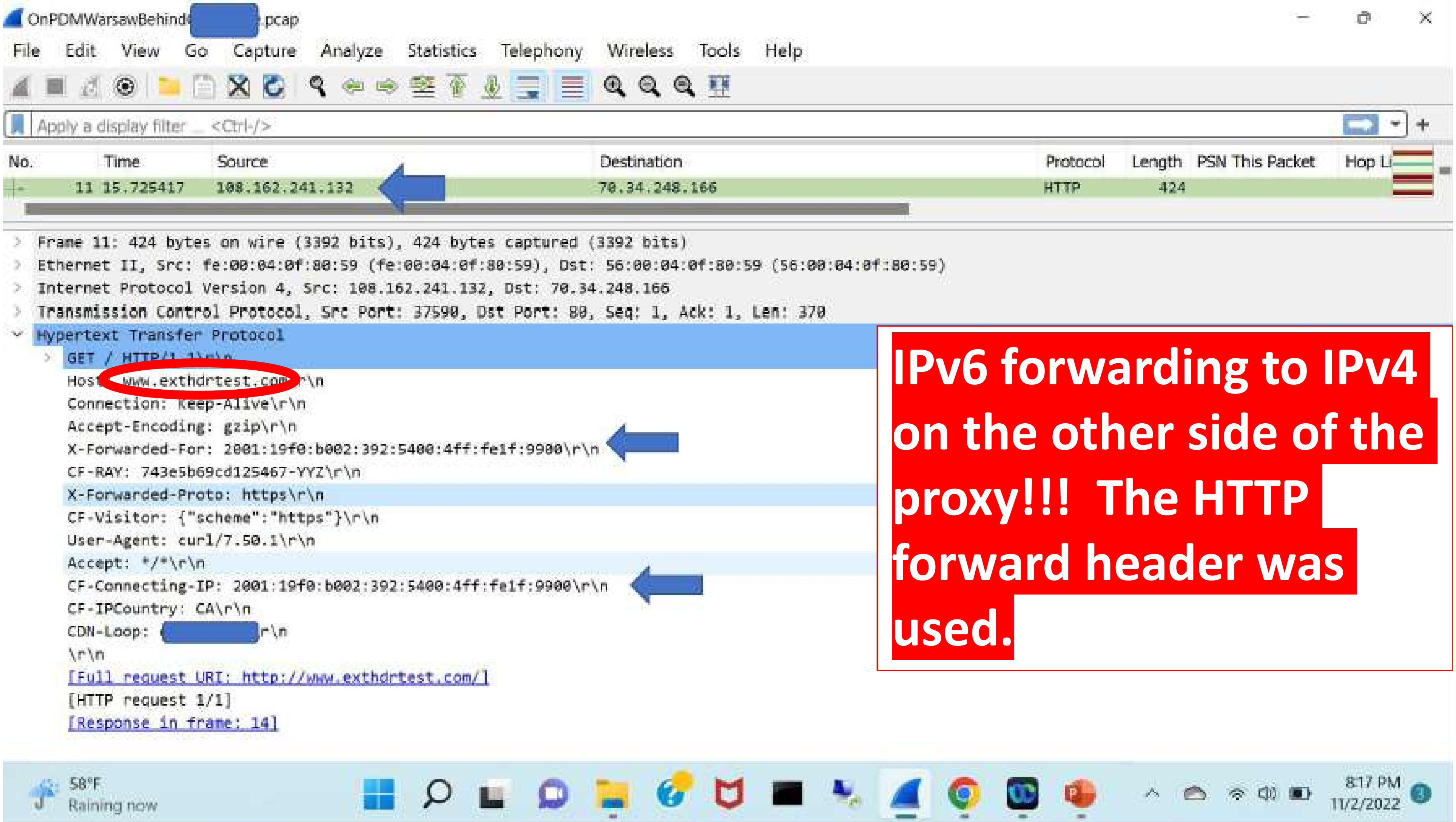
Type	Name	Content	Proxy status	TTL	Actions
A	exthdrtest.com	45.76.3.11	Proxied	Auto	Edit
⚠ A	ww4	45.76.3.11	DNS only	Auto	Edit
A	www	45.76.3.11	Proxied	Auto	Edit
⚠ AAAA	ww6	2001:19f0:5:3ce7:5400:4ff:fe31:1527	DNS only	Auto	Edit
AAAA	ww6p	2001:19f0:5:3ce7:5400:4ff:fe31:1527	Proxied	Auto	Edit
AAAA	www	2001:19f0:5:3ce7:5400:4ff:fe31:1527	Proxied	Auto	Edit
CNAME	autodiscover	adsredir.ionos.info	Proxied	Auto	Edit
CNAME	_domainconnect	_domainconnect.ionos.com	Proxied	Auto	Edit
CNAME	ww6cn	ww6.exthdrtest.com	DNS only	Auto	Edit
CNAME	ww6cnp	ww6p.exthdrtest.com	Proxied	Auto	Edit
MX	exthdrtest.com	mx00.ionos.com	10 DNS only	Auto	Edit
MX	exthdrtest.com	mx01.ionos.com	10 DNS only	Auto	Edit

So, the way many CDNs work is that they can either serve as “DNS only” or “DNS and Proxy”

Test #1: Going to Dual Stacked Web server and DNS



CDN is preferring
IPv4 if available!



OnPDMWarsawBehind[redacted].pcap
File Edit View Go Capture Analyze Statistics Telephony Wireless Tools Help
Apply a display filter ... <Ctrl-/>
No. Time Source Destination Protocol Length PSN This Packet Hop Li
11 15.725417 108.162.241.132 70.34.248.166 HTTP 424
Frame 11: 424 bytes on wire (3392 bits), 424 bytes captured (3392 bits)
Ethernet II, Src: fe:00:04:0f:80:59 (fe:00:04:0f:80:59), Dst: 56:00:04:0f:80:59 (56:00:04:0f:80:59)
Internet Protocol Version 4, Src: 108.162.241.132, Dst: 70.34.248.166
Transmission Control Protocol, Src Port: 37590, Dst Port: 80, Seq: 1, Ack: 1, Len: 370
Hypertext Transfer Protocol
Host: www.exthdrtest.com\r\n
Connection: Keep-Alive\r\n
Accept-Encoding: gzip\r\n
X-Forwarded-For: 2001:19f0:b002:392:5400:4ff:fe1f:9900\r\n
CF-RAY: 743e5b69cd125467-YYZ\r\n
X-Forwarded-Proto: https\r\n
CF-Visitor: {"scheme":"https"}\r\n
User-Agent: curl/7.50.1\r\n
Accept: */*\r\n
CF-Connecting-IP: 2001:19f0:b002:392:5400:4ff:fe1f:9900\r\n
CF-IPCountry: CA\r\n
CDN-Loop: [redacted]\r\n
\r\n
[Full request URI: http://www.exthdrtest.com/]
[HTTP request 1/1]
[Response in frame: 14]
IPv6 forwarding to IPv4 on the other side of the proxy!!! The HTTP forward header was used.
58°F
Raining now
8:17 PM
11/2/2022

Let’s take out the IPv4 definitions in DNS

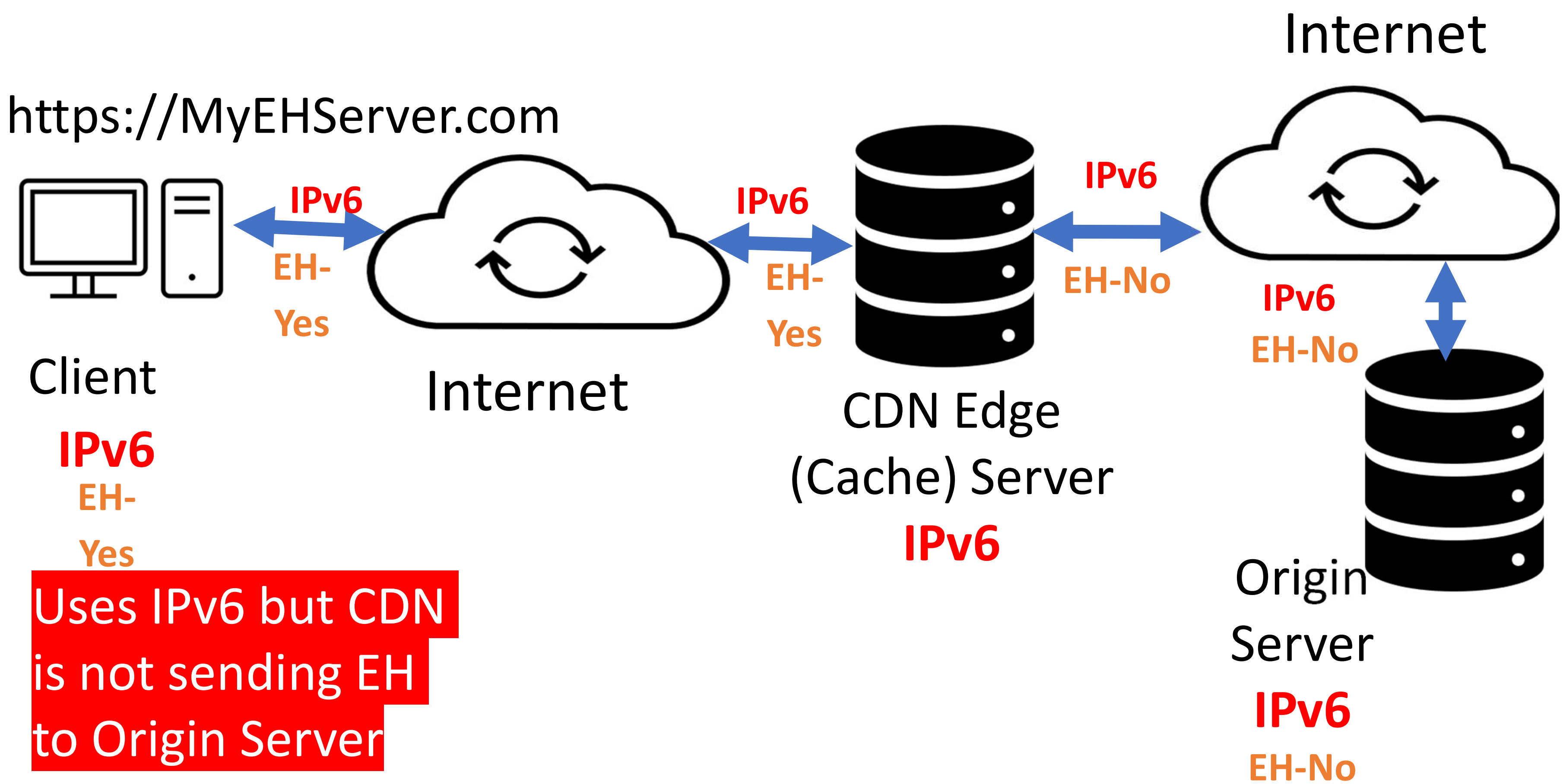
Original

;; A Records				
exthdrtest.com.	1	IN	A	45.76.3.11
ww4.exthdrtest.com.	1	IN	A	45.76.3.11
www.exthdrtest.com.	1	IN	A	45.76.3.11
;; AAAA Records				
• www.exthdrtest.com.	1	IN	AAAA	2001:19f0:5:3ce7:5400:4ff:fe31:1527

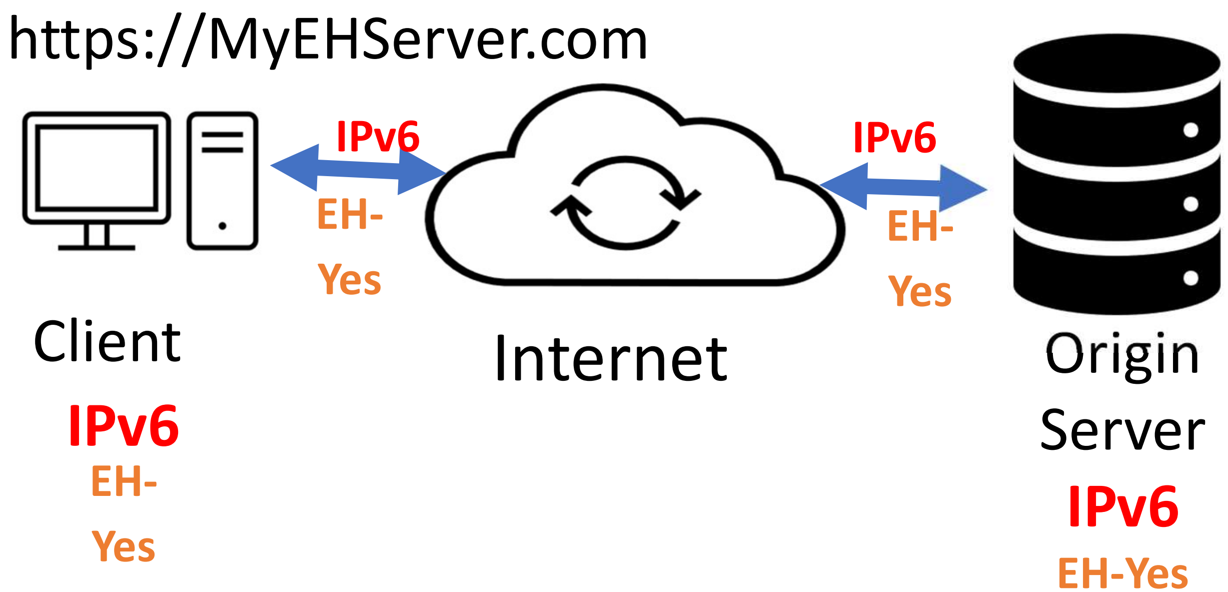
New

;; A Records				
ww4.exthdrtest.com.	**1**	**IN**	**A**	**45.76.3.11**
;; AAAA Records				
www.exthdrtest.com.	**1**	**IN**	**AAAA**	**2001:19f0:5:3ce7:5400:4ff:fe31:1527**

Test #2: IPv6-only Web Server and DNS AAAA only



Test #3: Doing DNS only at CDN



This works. We have managed to send EH to Origin Server by bypassing CDN Proxy. Now we are back to simple client / server scenario

Preliminary Conclusions

- Where EH can be sent with 90%+ probability (and why)
  - Standalone webservers (certain size / type EH)

- Where EH CANNOT be sent (to Origin Server) with 90%+ probability (and why)
  - CDN mediated web sites (unless in DNS-only mode)
  - “Proxy” may be the reason
  - More complications being researched

- What is unknown
  - Is it possible to collocate with CDN proxy to return EH?

Questions

- Should CDNs be encouraged to prioritize IPv6 over IPv4 in DNS?
- Should CDNs be encouraged to do IPv6 to Origin Server?
- How can EH be sent to Origin Server? (CDNs will not provide).

Students from NITK

National Institute of Technology
Karnataka, Surathkal

राष्ट्रीय प्रौद्योगिकी संस्थान
कर्नाटक, सुरत्कल

Chinmaya Sharma,
Amogh Umesh,
Balajinaidu V.

Professor: Dr. Mohit Tahiliani

Implementation of PDMv2 in eBPF

Balajinaidu V
Amogh Umesh
Chinmaya Sharma

tc-BPF

- Subset of eBPF programs attached at qdisc level
- Can be attached to both ingress and egress compared to only ingress in XDP
- Better packet mangling capability
- Executed after sk_buff is created
- Not good for complete packet rewrites
- Doesn’t require hardware changes

~~Impl~~ementation of PDM using eBPF

- Easy development and testing.
- Using tc-BPF, so that we can attach to both ingress and egress of a interface.
- Modifying the packet after sk_buff is constructed
- eBPF maps to store the 5-tuple state.

Progress till now

- Explored using eBPF to add Extension Headers to packets
- Implemented PDM using tc-BPF
- Implementation uses eBPF programs attached to qdisc at both ingress and egress
- Uses BPF maps to store information specific to a flow
- Currently in the process of implementing PDMv2 in eBPF

Challenges for encryption in eBPF

- Lack of library support for encryption in eBPF
- Limit on the number of instructions
- Lack of resources of other implementations
- eBPF verifier causing errors in loading in various situations
- May not be efficient due to eBPF and encryption being CPU intensive

~~Test~~ing with XOR Encryption

- We tested XOR encryption in eBPF to check if it could be done without any problems
- Faced with an error of too many processing instructions from the verifier
- Error due to setting the loop iterative variable as 64 bits unsigned integer instead of 32 bits, which may have caused the verifier to check for more states
- Successfully encrypted PDM in eBPF using XOR encryption

Challenges with verifier

- Verifier checks are extremely stringent
- Checks all possible outputs of function to verify its termination
- This may get out of hand if manipulating large data
- Difficult for encryption

Some useful Links

ebpf.io

RFC8250

PDMv2-Draft

tc-BPF

Next steps …

- Test with:
  - CDNs (continued / collocation)
  - Cloud providers
  - Routers
  - ISPs
  - Load balancers
  - OSs

- Need to test ALL extension headers!

- This will be a long process!

Questions?

Contact:

info@iiesoc.in

president@industrynetcouncil.org